

Amendments to the Claims:

This listing of claims replaces all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (previously presented) A system, comprising:

a memory configured to store data associated with a plurality of incoming streams of different speeds;

an interface controller comprising a first arbitration element to arbitrate among the streams to store the data in the memory, the first arbitration element including a number of first entries, one of the first entries indicating which of the streams is to be serviced in a particular first time slot, the streams being assigned to the first entries based on the speeds of the streams; and

a dispatch unit comprising a second arbitration element to arbitrate among the streams to read the data from the memory, the second arbitration element including a number of second entries, one of the second entries indicating which of the streams is to be serviced in a particular second time slot, the streams being assigned to the second entries based on the speeds of the streams.

2. (previously presented) The system of claim 1, where the memory

includes:

a plurality of memory buckets corresponding to the streams.

3. (previously presented) The system of claim 2, where the memory buckets have a fixed size.

4. (previously presented) The system of claim 1, each of the first entries including a stream number that identifies one of the streams.

5. (previously presented) The system of claim 1, where the number of the first entries in the first arbitration element is programmable.

6. (previously presented) The system of claim 4, where the interface controller is configured to:

- read one of the stream numbers from the first arbitration element,
- gather data corresponding to the identified stream, and
- transfer the data to the memory.

7. (previously presented) The system of claim 6, where the interface controller is further configured to send a stream identifier with the data transferred to the memory.

8. (previously presented) The system of claim 7, where the memory is further configured to sort the data from the interface controller based on the stream identifier.

9. (previously presented) The system of claim 1, where the first and second arbitration elements are synchronized.

10. (previously presented) The system of claim 1, each of the second entries including a stream number that identifies one of the streams.

11. (previously presented) The system of claim 1, where the number of the second entries in the second arbitration element is programmable.

12. (previously presented) The system of claim 10, where the dispatch unit is configured to:

read one of the stream numbers from the second arbitration element,
read data corresponding to the identified stream from the memory, and
output the data for processing.

13. (original) The system of claim 1, further comprising:
flow control logic configured to initiate flow control on the storing of data in the memory.

14. (previously presented) The system of claim 13, where the flow control includes dropping data from the stream.

15. (previously presented) The system of claim 13, where the flow control includes causing the interface controller to stop storing data from the stream in the memory.

16. (previously presented) The system of claim 13, where the flow control logic includes:

a buffer configured to temporarily store the data from the interface controller in a plurality of entries,

a counter configured to determine a number of entries in the buffer corresponding to each of the streams, and

a comparator configured to determine whether to initiate the flow control for each of the streams based on the determined number of entries for the stream.

17. (previously presented) The system of claim 16, where the comparator is configured to compare the determined number of entries for a stream to a watermark and initiate the flow control for the stream when the determined number of entries exceeds the watermark.

18. (previously presented) The system of claim 17, where the comparator is further configured to compare the determined number of entries for the stream to a

second watermark and drop data from the stream when the determined number of entries exceeds the second watermark.

19. (previously presented) The system of claim 1, where each of the streams has an associated watermark for performing flow control on the storing of data in the memory.

20. (previously presented) The system of claim 1, where each of the streams has two associated watermarks for performing flow control on the storing of data in the memory.

21. (previously presented) A method, comprising:
storing data from a plurality of streams of potentially different speeds in a memory using a first arbitration scheme that stores data associated with a faster one of the streams in the memory at a higher rate than data associated with a slower one of the streams; and

reading the data from the memory using a second arbitration scheme that reads the data associated with the faster one of the streams from the memory at a higher rate than the data associated with the slower one of the streams.

22. (previously presented) The method of claim 21, where the storing includes:

storing the data in a plurality of memory buckets in the memory based on the streams to which the data belongs.

23. (previously presented) The method of claim 21, further comprising:
writing a plurality of entries into a first arbitration element that implements the first arbitration scheme, where each of the entries including a stream number that identifies one of the streams.

24. (previously presented) The method of claim 23, where the number of the entries in the first arbitration element for the faster one of the streams being greater than the number of the entries in the first arbitration element for the slower one of the streams.

25. (previously presented) The method of claim 23, where the storing includes:

reading one of the stream numbers from the first arbitration element,
gathering data corresponding to the identified stream, and
transferring the data to the memory.

26. (previously presented) The method of claim 25, where the transferring includes:

sending a stream identifier with the data transferred to the memory.

27. (previously presented) The method of claim 26, where the storing further includes:

sorting the data based on the stream identifier.

28. (previously presented) The method of claim 21, where the first and second arbitration elements are synchronized.

29. (previously presented) The method of claim 21, further comprising:
writing a plurality of entries into a second arbitration element that implements the second arbitration scheme, each of the entries including a stream number that identifies one of the streams.

30. (previously presented) The method of claim 29, where the number of the entries in the second arbitration element for the faster one of the streams being greater than the number of the entries in the second arbitration element for the slower one of the streams.

31. (previously presented) The method of claim 29, where the reading includes:

obtaining one of the stream numbers from the second arbitration element,
obtaining data corresponding to the identified stream from the memory, and
outputting the data for processing.

32. (original) The method of claim 21, further comprising:
initiating flow control on the storing of data in the memory.
33. (previously presented) The method of claim 32, where the initiating includes:
temporarily storing the data in a plurality of entries in a buffer,
determining a number of entries in the buffer corresponding to each of the streams, and
determining whether to initiate the flow control for each of the streams based on the determined number of entries for the stream.
34. (previously presented) The method of claim 33, where the determining whether to initiate the flow control includes:
comparing the determined number of entries for a stream to a watermark, and
initiating the flow control for the stream when the determined number of entries exceeds the watermark.
35. (previously presented) The method of claim 34, where the determining whether to initiate the flow control includes:
comparing the determined number of entries for the stream to a second watermark, and

dropping data from the stream when the determined number of entries exceeds the second watermark.

36. (previously presented) The method of claim 32, where the initiating the flow control includes:

dropping data from the stream.

37. (previously presented) The method of claim 32, where the initiating the flow control includes:

stopping the storing of data from the stream in the memory.

38. (previously presented) The method of claim 21, where each of the streams has an associated watermark for performing flow control for the associated stream.

39. (previously presented) The method of claim 21, where each of the streams has two associated watermarks for performing flow control for the associated stream.

40. (currently amended) A system for performing flow control on data in a plurality of incoming streams of variable speeds, comprising:

a buffer configured to temporarily store data from a plurality of streams of variable speeds in a plurality of entries;

a counter configured to determine a number of entries in the buffer corresponding to each of the streams; and

a comparator configured to compare the determined number of entries, for one of the streams, to a watermark and determine whether to initiate flow control for ~~[[each]]~~ the one of the streams based on a result of the comparison, where the watermark is particular to the one of the streams and independent of another watermark set for another one of the streams ~~the determined number of entries for the stream.~~

41. (currently amended) The system of claim 40, where the comparator is configured to compare the determined number of entries for ~~a stream~~ the one of the streams to ~~[[a]]~~ the watermark and initiate the flow control for the ~~stream~~ one of the streams when the determined number of entries exceeds the watermark.

42. (currently amended) The system of claim 41, where the flow control includes dropping data from the ~~stream~~ one of the streams.

43. (currently amended) The system of claim 41, where the flow control includes causing a sender of the data to stop outputting data in the ~~stream~~ one of the streams.

44. (currently amended) The system of claim 41, where the comparator is further configured to compare the determined number of entries for the ~~stream~~ one of the

streams to a second watermark and drop data from the ~~stream~~ one of the streams when the determined number of entries exceeds the second watermark.

45. (canceled)

46. (previously presented) The system of claim 40, where each of the streams has two associated watermarks for use in performing flow control on the associated stream.

47. (original) The system of claim 40, further comprising:
a memory configured to store data from the buffer;
an interface controller having a first arbitration element and configured to output the data to the buffer using the first arbitration element; and
a dispatch unit having a second arbitration element and configured to read the data from the memory using the second arbitration element.

48. (original) A method for performing flow control on data in a plurality of incoming streams of variable speeds, comprising:
storing data from a plurality of streams of variable speeds in a plurality of entries of a buffer;
determining a number of entries in the buffer corresponding to each of the streams; and

determining whether to initiate the flow control for each of the streams based on the determined number of entries for the stream.

49. (previously presented) The method of claim 48, where the determining whether to initiate the flow control includes:

comparing the determined number of entries for a stream to a watermark, and
initiating the flow control for the stream when the determined number of entries exceeds the watermark.

50. (previously presented) The method of claim 49, where the initiating the flow control includes:

dropping data from the stream.

51. (previously presented) The method of claim 49, where the initiating the flow control includes:

causing a sender of the data to stop outputting data in the stream.

52. (previously presented) The method of claim 49, where the determining whether to initiate the flow control includes:

comparing the determined number of entries for the stream to a second watermark, and

dropping data from the stream when the determined number of entries exceeds the second watermark.

53. (previously presented) The method of claim 49, where each of the streams has an associated watermark.

54. (previously presented) The method of claim 49, where each of the streams has two associated watermarks.

55. (original) A system for performing flow control on data in a plurality of incoming streams of variable speeds, comprising:

a buffer configured to temporarily store data from a plurality of streams of variable speeds in a plurality of entries;

a counter configured to determine a number of entries in the buffer corresponding to each of the streams; and

a comparator configured to:

compare the determined number of entries for a stream to first and second watermarks,

initiate flow control for the stream when the determined number of entries exceeds the first watermark, and

drop data from the stream when the determined number of entries exceeds the second watermark.

56. (previously presented) A network device, comprising:
an input interface configured to:
 receive a plurality of packets belonging to a plurality of streams of
differing speeds,
 access a first arbitration scheme that services a faster one of the streams
more often than a slower one of the streams, and
 output the packets based on the first arbitration scheme;
input logic comprising:
 flow control logic configured to initiate flow control on the packets output
by the input interface,
 a memory configured to store the packets from the input interface, and
 a dispatch unit configured to:
 access a second arbitration scheme that services the faster one of
the streams more often than the slower one of the streams, and
 read the packets from the memory based on the second arbitration
scheme; and
one or more packet processors configured to process the packets from the dispatch
unit.

57. (previously presented) A network device, comprising:

means for receiving a plurality of packets belonging to a plurality of streams of potentially different speeds;

means for storing the packets based on a first arbitration scheme that stores the packets based on the speeds of the streams to which the packets belong;

means for performing flow control on the storing of the packets;

means for reading the packets based on a second arbitration scheme that reads the packets based on the speeds of the streams to which the packets belong; and

means for processing the packets read based on the second arbitration scheme.

58. (canceled)

59. (previously presented) The system of claim 1, where at least one of the first arbitration element or the second arbitration element is configured to be reprogrammed when the speed of one of the streams changes.

60. (canceled)

61. (previously presented) The method of claim 21, further comprising:
receiving an input that the speed of one of the streams has changed; and
reprogramming at least one of the first arbitration element or the second arbitration element based on the received input.

62. (previously presented) The system of claim 59, where the at least one of the first arbitration element or the second arbitration element is reprogrammed to change the number of the first or second entries assigned to the one of the streams.

63. (previously presented) The method of claim 61, where reprogramming the at least one of the first arbitration element or the second arbitration element includes changing the rate at which data associated with the one of the streams is stored in or read from the memory.

64. (new) The system of claim 55, where the first and second watermarks for one of the streams is independent of the first and second watermarks for another one of the streams.